

Reconcile Users' Guide

John H. Howard

TR98-04a December 1998

Abstract

Reconcile combines different versions of file directories, propagating all updates between them and making them identical but never losing updates at one site because of updates performed concurrently at another. Among its applications are: - moving files to and from a laptop This report describes the program in detail, including motivation, basic concepts, applications, and the program's interface.

This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Research Laboratories, Inc.; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Research Laboratories, Inc. All rights reserved.

Reconcile Users' Guide

John H Howard

Abstract

Reconcile combines different versions of file directories, propagating all updates between them and making them identical but never losing updates at one site because of updates performed concurrently at another. Among its applications are:

- moving files to and from a laptop
- immunity from file server failures
- peer-to-peer file sharing without a file server
- making private backups on a removable disk
- replicating files between geographically separated file systems

This report describes the program in detail, including motivation, basic concepts, applications, and the program's interface.

Copyright © Mitsubishi Electric Information Technology Center America, 1993-1998
201 Broadway, Cambridge, Massachusetts 02139

This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Information Technology Center America, of Cambridge, Massachusetts; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Information Technology Center America. All rights reserved.

CONTENTS

QUICK START	1
Installation	1
First time setup.....	1
Normal operation	1
Ignoring selected files	1
Beta Test	1
 OVERVIEW	 2
Features	2
Sites, journals, and versions	2
 OPERATION	 3
Read and update individual journals	3
Merge journals	4
Propagate missing actions	4
Purge obsolete versions.....	4
Write updated journals	5
Interactive review (optional).....	5
 EXAMPLES	 5
Creating a new journal.....	5
Making private backups on a removable disk.....	5
Local copy from a file server.....	6
Moving files to and from a laptop	6
Server-less file sharing.....	7
 REFERENCE MATERIAL.....	 8
Command Reference.....	8
Syntax.....	8
Options:	8
Modes for individual sites:	8
Site identification:	8
Setting the time.....	8
Journal file format	9
Header line:	9
Version line:	9
Known sites:	10
Files used.....	10
Special directives.....	10
#ignore <pattern>	11
#alwaysnew <pattern>.....	11
#deletedirs <pattern>.....	11
#notranslate <pattern>.....	11
#normal <pattern>	11

Quick Start

Installation

√ Copy the reconcile program file to an appropriate directory, such as /usr/local/bin (for Unix) or c:/windows (for Windows 95.)

First time setup

√ Make sure your disks have unique, descriptive labels using either the DOS label command or the Windows disk property dialog.

√ Create a brand-new journal for existing directory /homes/zymurgy with the command:

```
reconcile -v -c /homes/zymurgy
```

(substitute the directory name of your choice for “zymurgy”; but you can’t use “.”). This will take a while, as reconcile computes digests of every new file it finds.

√ Create a copy of the directory and journal somewhere else, for example on d:/zymurgy:

```
mkdir d:/zymurgy
reconcile -v -c d:/zymurgy
reconcile -v /homes/zymurgy d:/zymurgy
```

Note that the first two commands simply create an empty directory and journal. The third one fills the directory by copying from /homes/zymurgy.

Normal operation

To reconcile /homes/zymurgy and d:/zymurgy with each other, use the command:

```
reconcile -v /homes/zymurgy d:/zymurgy
```

You may wish to put this command in a batch file or on an icon in your user interface.

Ignoring selected files

To prevent reconcile from copying some sorts of files, for example core dumps or object files on Unix, create a file named “.reconcilerc” containing directives like:

```
#ignore *.o
#ignore core
```

Beta Test

Your feedback is important to me, including both suggestions and bug reports. The trace file (/tmp/reconcile.log) gives a record of what reconcile did; please make sure to send a copy to me along with any reports, even if you’re not sure it’s a bug or just a mis-feature.

To be extra careful after installing a new version, do a run with the -n option:

```
reconcile -n -v /homes/zymurgy d:/zymurgy
```

This will tell you what reconcile would do, without actually doing it.

PLEASE READ THE REST OF THIS MANUAL! It contains valuable information about what to expect from reconcile and how to use its features.

Overview

Reconcile compares file directories, finds the differences, and updates the directories so that they all contain exactly the same files. It does this safely and automatically, requiring user intervention only when there are unavoidable conflicts between the versions of files stored in different locations.

Features

Safe updating is accomplished by using a journal of past updates for each directory. An older version of a file is deleted only if the journals indicate that the newer version was derived directly from the older one. If this criterion is not met, the older version is renamed, rather than being deleted. This allows the user to inspect the two files and determine how to merge or otherwise resolve the conflict between them. Reconcile prints a warning message every time it runs, until the renamed file is dealt with.

A newer version of a file can safely replace an older one whenever a journal contains both of them. This can only happen if the newer version was created by a deliberate act of the user or some application program, replacing the older version. The journaling mechanism also handles deletions by noting the disappearance of files as well as the appearance of new versions. This means that if a file is deleted at any site, Reconcile can safely delete it at other sites. This is a key advantage over programs which simply compare time-stamps on files. It also avoids relying on synchronized clocks between computers.

Reconcile is *automatic* in the sense that it does not require user intervention. It can be invoked from a batch file, periodically through a system scheduler agent, when the user logs in or logs out, or at other convenient moments. In its automatic mode, Reconcile runs to completion, logging messages about any conflicts between versions of files or other problems it encounters. In the unusual case of a conflict that Reconcile can not resolve by itself, it renames one of the conflicting version with a distinctive name to make manual resolution easier.

In addition to its automatic mode, Reconcile can be run *interactively*. In its interactive mode it locates all unresolved conflicts and other problems and offers the user an opportunity to correct them manually.

Reconcile can also be told to ignore certain files, such as backup copies and object files generated by compilers. This avoids unnecessary conflicts between files which are derived from other files rather than being created or updated directly by user actions.

Finally, Reconcile offers *cross-platform compatibility*. Versions exist both for Windows and for Unix. When copying text-mode files (defined as files containing only white space and printable characters) between Windows and Unix sites, Reconcile automatically converts line endings to the form used by the target site.

Sites, journals, and versions

A **site** is one version or copy of a set of files, organized as a subtree of the file system hierarchy. The basic purpose of Reconcile is to combine sites, making them all the same by safely copying individual files between them. A typical site might be an individual's

working directory on a particular computer. If you use two computers, you would have a site on each of them.

One should not think of a site as being the total disk storage on any one computer. Computers contain many unrelated directories reflecting how it is configured. A personal computer, for example, might contain separate directories for system software, installed applications, projects, and individuals' working data. While file systems typically combine these into a single hierarchy, it is easier to think of them as being separate.

A site may be stored on a portable medium such as a floppy diskette or Zip disk, or on a network file server, as well as on the internal disk of a computer. The only real limitation is that the site appear somewhere within the file system. All files and subdirectories beneath the site's top-level directory are part of the site. If the file system supports symbolic links, the links are included in the site but not the objects they refer to.

To do its work, Reconcile creates or updates a **journal** file for each site, listing the history and status of every file at the site. A journal, in turn, is a list of successive versions of files contained in a directory. A journal also contains subjournals of the subdirectories found. A new **version** of a file is created every time the contents of the file are modified, or if the file is deleted. Thus a version describes both an action, which can be either "update" or "delete", and the contents of the file. By including deletion operations in journals Reconcile can (safely) propagate deletions to other sites, again checking for conflicts.

Journal entries for versions of files contain the action, file name, a **time-stamp** (the date and time the file was most recently modified) and a **digest**, which is a checksum of the entire file computed in such a way that it is extremely unlikely that two different files would ever have the same digest. In order to promote portability between UNIX, Windows, and Macintosh computers, Reconcile ignores differences in line endings (CR, LF, or CRLF) when computing digests of text files, and (optionally) translates line endings when copying text files between systems of different type.

Reconcile uses the time-stamp only to determine when it should compute a new digest. It does not depend on times being synchronized between computers, and is even insensitive to arbitrary changes in the clock on one computer. (Since time stamps have a resolution of a second or two, it is extremely unlikely that two different versions of the same file would have the same timestamp.)

It might be a good idea to include other file information, such as protection flags, in the journal, but this is currently not done in Reconcile because of the lack of consistency between Windows and UNIX in this area.

Operation

Reconcile operates in several phases sequentially:

Read and update individual journals

After reading the journal for each subdirectory, Reconcile scans the current contents of the actual subdirectory, comparing the files actually present with the files listed in the

journal. Create new journal entries reflecting creation, modification, and deletion of files at each site. Note that a deletion can be inferred from the absence of an actual file for which a journal entry exists.

Merge journals

Reconcile merges the journal entries for each file by finding a maximum matching subsequence of entries among all the sites. The main purpose of this is to find the last common entry for each file (that is, the most recent matching entry among all sites.) Therefore the merging operation gives much more weight to matching the most recent version of each file in the journal.

Propagate missing actions

If the last matching entry of a file is present at all sites, then they are already consistent and no action is needed.

- If one site has one or more unique entries after the last common entry, but the others do not, the missing unique entries are replicated to the other site by copying or deleting files as required. (The existence of the common entry is definitive evidence that the new entries represent intentional creation, modification, or deletion of the file.)
- If each site has different entries after the last common entry, there is a conflict. In this case, the older versions are renamed rather than being deleted, using names of the form “filename#1.extension”. The number is chosen to generate a unique filename. This feature can be turned off with the `#alwaysnew` directive.
- For safety’s sake, deleted subdirectories are also renamed rather than being deleted, using the same naming conventions. This feature can be turned off with the `#deletedirs` directive.

Whenever a file or directory is renamed due to conflict or directory deletion, a reminder is left in the journal for future review. Presence of the unusual `#` character also makes it easy to find such saved files and directories.

Purge obsolete versions

Old journal entries become obsolete when they are known at all sites. Reconcile tracks this by maintaining a cumulative list of all sites being reconciled now and other sites that have participated in past reconciliations with other known sites. It marks each version with the set of known sites which have heard about the version. Once a newer version is known at all sites where the older version is known, the older version is no longer needed and is purged from the journal.

If a known site has not participated in a reconciliation for a very long time, old versions will accumulate and the journal will slowly grow. To deal with this, Reconcile abandons known sites which have not been heard from for a month or more. (It generates warning messages for several runs before doing this.) If an abandoned site rejoins the set of known sites, its old versions of files will be treated as conflicting versions, generating false warnings, rather than being updated cleanly.

There is a special case for old deletion entries in the journal. If some, but not all, sites purge an old deletion, the others will see this as a missing journal entry the next time they reconcile and therefore restore the purged version entry. To deal with this, reconcile discards month-old deletions regardless of whether they are known at all sites.

Write updated journals

After updating, Reconcile writes out the updated journals at all sites. In order to minimize the possibility of a crash leaving a partial journal written, Reconcile actually writes the new journal with extension “.jnw”, then renames it to “.jnl” once the write is complete. Reconcile also saves the previous copy of the journal by renaming it with extension “.jbk”.

Interactive review (optional)

Review the journals for unresolved conflicts, either from this run or from a previous one. For each conflict, display the old and new file names, ask the user how to resolve the conflict, and perform the action the user specifies; similarly for deleted directories. This affects the file system only; the journals will be updated by the next run.

Examples

This section describes how to use Reconcile in several common scenarios: saving to a removable disk, moving files between computers, file sharing in a network, and server-less file sharing.

Creating a new journal

To create a new journal for existing files in c:\work.jnl, use the command:

```
reconcile -w c:\work.jnl
```

It is important that you label each disk (both fixed and removable) with a unique and meaningful name. Use the label command to review and set the label of any disk before you create a new directory on it. For example, if your office computer is named "Thor", you might label its C disk with the command:

```
label c: THOR-C
```

Making private backups on a removable disk

Suppose you keep your working files in the directory c:\work, and want to keep a backup copy on a removable disk in drive z. Create a journal for c:\work as described above. Also label the removable disk, create an empty directory and journal on it:

```
label z: WORKCOPY
mkdir z:\work
reconcile -w z:\work.jnl
```

You only need to do the above steps once.

Once things are set up, you can reconcile the two sites by running the command

```
reconcile c:\work z:\work
```

Reconcile generates messages describing any actions it takes and problems it finds. You don't need to remember these messages on the spot, as any important warnings will be

repeated until you do something about them. The first run may take a little while as many files are copied and the journal initialized, but subsequent runs will be much faster.

A restriction in this example is that the removable disk must be large enough to hold all the files. If you use a Zip disk this probably won't be a problem.

Local copy from a file server

The example above used a removable disk, but could equally well use a disk shared across a network by another computer. The only change is to replace the removable disk's drive letter (z: in the example) with the corresponding name of the network drive or subdirectory you use. Thus, if the server's copy happened to be mounted as h:\homes\me, the reconciliation command would be:

```
reconcile c:\work h:\homes\me
```

Since the server is usually available, you can schedule periodic automatic reconciliations, preferably at a time when nobody is likely to be using the files involved. You can occasionally use the interactive mode to review for problems, using the command

```
reconcile -i c:\work
```

Although in fact there are two complete copies of the files involved, one can think of this as offering a single replicated set of files. The cost is the extra disk storage needed to store copies of all the files, and the benefits are availability and performance. Availability is obvious - you don't need the network or file server to work on the local copy.

Performance comes from the fact that your computer doesn't need to be in continuous communication with the file server, telling it about every file it updates and checking for updates performed elsewhere. The only performance cost is the time taken to do the periodic reconciliation, typically late at night when the computer and network are idle.

Moving files to and from a laptop

Now suppose you occasionally use a laptop or other second computer, and would like use your files there. You can do this using any medium accessible to both the laptop and the main computer. This might be a removable disk or it might be a file server; the example below uses the latter.

It is be convenient to use the same working directory name (c:\work) on the laptop as you do on your main computer. In order to guarantee that the sites have unique names, you should label the laptop's hard disk uniquely. Then create an empty directory and journal:

```
(first time only, on the laptop)
label c: LAPTOP
mkdir c:\work
reconcile -w c:\work.jnl
```

Now you're ready to go. To bring the laptop up to date:

```
(on the main computer)
reconcile c:\work h:\homes\me

(on the laptop)
reconcile c:\work h:\homes\me
```

Notice that you use exactly the same command on both computers, assuming that the local directory names are the same. The first time, Reconcile will notice that the laptop's c:\work is empty and proceed to copy all files from the file server to it. Thereafter it will only copy updated files.

When you're done with the laptop and want to return to the main computer, simply reverse the process:

```
(on the laptop)
reconcile c:\work h:\homes\me

(on the main computer)
reconcile c:\work h:\homes\me
```

From now on, you can use either computer at will, provided that you remember to run Reconcile at the beginning and end of each working session. At the beginning, the reconciliation will copy the latest files from the file server to the hard disk, and the ending reconciliation will bring the file server up to date.

Suppose you do happen to forget to run Reconcile at the beginning or end of a session, and you update some file (say oops.doc) separately on both computers. The next time you run Reconcile, you will get the error message:

```
WARNING: oops#1.doc is a saved version of oops.doc
```

At this point you will probably want to look at both files to find the differences between them and to copy any valuable changes from oops#1.doc into oops.doc. Different applications have different ways to do this: for text files there is a convenient tool named diff or windiff which finds and displays a minimum set of differences. If all else fails you can simply display or print the files and compare them manually. When you are satisfied that you have recovered everything into oops.doc, delete oops#1.doc.

It is OK if you do not resolve the conflict right away. Reconcile keeps a record of unresolved conflicts in its journal file and warns you every time it runs, until you delete or rename the saved file (oops#1.doc.) It's also easy to locate such files by searching for file names containing the # character, which is unusual in normal file names.

Server-less file sharing

Both Windows and Unix allow client computers to act as servers, offering other computers access to their local disks. This can be done using a variety of interconnection techniques including not only networks but also modems and even printer cables. With any such connection, the client can perform a reconciliation against the server. This is convenient for a portable computer; it even works across a dial-in connection provided that the amount of updated information is reasonable. In the general case one can imagine a very loosely coupled network of portable computers which occasionally meet and exchange information without ever needing a central repository.

Reference Material

Command Reference

Syntax

```
reconcile [options] [[mode] [-j journal] site[.jnl]] ...
```

Options:

- h print a description of the command syntax, don't do anything else.
- n do not update anything; just say what would happen
- p pause and wait for dialog input if any errors encountered (Windows only)
- pn pause for *n* (decimal) seconds before exiting (Windows only)
- q quiet – suppress updating messages
- v verbose – display progress messages
- d "*directive*" apply *directive* throughout unless overridden
- l *tracefile* log actions performed to *tracefile*
- t *tracefile* trace details about actions performed to *tracefile*

Modes for individual sites:

- a abandon this site, that is, forget all about it.
- c create a new journal (actually this is the same as -w)
- w don't read the journal ("write-only")
- r don't write the journal ("read-only")
- o do not update the site ("only journal")
- s use this site as a time reference
- j *journal* identifies the journal file, overriding the rules given below.

Note that a new mode is established for each different site.

Site identification:

- If site names a directory, the journal file is derived from it according to any one of the following cases, decomposing it into path/name:
 1. path/name/name.jnl (this is the preferred case)
 2. path/name.jnl (obtained by replacing any extension)
- If site.jnl names a file, it is the journal file. The directory name is obtained by stripping off the extension (.jnl in this example). This corresponds to case 2 above.
- For cross-platform compatibility, both / and \ are accepted as path separators. Upper- or lower-case is significant only if it is for the operating system.
- Disk letters (or for Unix, mount points) may be replaced by labels in order to derive unambiguous names. Thus if the C disk is labeled MYCOMPUTER, Reconcile treats C:\homes and MYCOMPUTER:\homes as equivalent.

It is important that you label each disk (both fixed and removable) with a unique and meaningful name.

Setting the time

The first accessible, writable site with the -s flag, if any, is used as a time reference. The time at that site is determined by creating a dummy file and reading its timestamp. If the time thus obtained differs from the local clock by more than a few seconds, reconcile

attempts to reset the local clock to agree with the reference site. Note that on Unix this requires root privileges and on Windows NT it requires the “Change the system time” privilege.

Journal file format

Journal files are standard text files that can be observed (and even changed, at your own risk) with any text editor. The files contain a header line, several lines identifying known sites, and then one line for each version of each file.

Header line:

```
Journal of <sitename> (<systype>) - <programname>
```

where <sitename> is the fully qualified name of the file hierarchy, <systype> is the system type (DOS or UNIX at present), and <programname> gives the name and version of the Reconcile program that wrote the journal.

Version line:

Version lines identify particular versions of files in the format:

```
<verb> <date> <time>Z <name><t> ?mask dt=<digest> <remarks>
```

<verb> is one of a limited set of symbols denoting possible actions:

```
+      create or update, making a new version
-      delete
```

(the following three lines are only generated in -o mode)

```
>      another site has a more recent version
<      another site has a more recent deletion
!      another site has a more recent version,
        but it conflicts with the previous version at this site.
```

<date> is the date the action occurred (yy/mm/dd format)

<time>Z is the time the action occurred (hh:mm:ss format)
(Z indicates Coordinated Universal Time)

<name> is the file's name, followed directly by the type (no intervening white space)

<t> is a single type character:

```
(nothing) for an ordinary file
/ for a subdirectory
@ for a symbolic link
```

?mask is a bit-mask indicating which other sites do *not* know about this version yet. The individual mask bits are defined in the known site (\$) lines.

dt=<digest> gives a hash code, or digest, of the contents of the file, used to determine if files are identical even though they might have different dates. “dt” means the file looks like a text file, “db” is used for binary files.

<remarks> are arbitrary comments. Saved files are indicated by the special remark:

```
!was <original name>
```

and deleted directories by the special remark:

```
!deleted <original name>
```

Known sites:

The known site lines give the date and time of the most recent reconciliation for each known site. Each line is of the form:

```
$ <date> <time> <sitename> ?mask
```

The date and time give the last time the site has been heard from, and the hexadecimal mask defines the mask bit used for this site in version lines.

Files used

The following files may be used by Reconcile.

<site>.jnl - journal of the contents of directory hierarchy <site>

<site>.jnw - new the journal being written (exists temporarily only)

<site>.jbk - previous copy of <site>.jnl

.reconcilerc - special directives applying to selected subdirectories

<filename>#nnn.<ext> - saved conflicting version of file <filename>.<ext>

<dirname>#nnn - saved version of deleted directory <dirname>

/tmp/reconcile.log and /tmp/reconcile.oldlog - default name for trace of actions performed, if not overridden by -l or -t option. The trace file contains the various update and error messages generated during reconciliation. If the -t option is used, it also contains detailed information about why reconcile performed the actions.

Special directives

Any directory or subdirectory may contain a file named “.reconcilerc”, containing directives which modify processing of that directory and its subdirectories. Comment lines starting with an asterisk are also accepted. Directives apply to matching files in the directory in which they appear and also all subdirectories of that directory.

Directives may also be edited into a journal file, in which case they apply to the current directory or subdirectory defined by the most recent header line. Top level directives can also be provided by the -d command option.

Patterns, indicated by <pattern>, are templates for file names in which wild card ("*") characters match arbitrary sequences of zero or more characters (other than path separators). Patterns or parts of patterns may be enclosed in double quotes; a pair of double quote characters (within a quoted pattern) represents one double quote character. All other characters match literally. Patterns must match the entire file name, so to ignore all files whose names contain the letters “foo” you would use *foo*. If several patterns match a file name, the first matching one is used.

Reconcile accepts the following directives:

#ignore <pattern>

causes matching files to be ignored during reconciliation. Reconcile creates implicit “#ignore *.jnl” and “#ignore *.jbk” directives at the root of the directory hierarchy. Note that the .reconcilerc file itself is *not* automatically ignored but is propagated normally.

#alwaysnew <pattern>

tells reconcile always to copy the newer version of a file, without preserving the older version even in the case of a detected conflict. *Use with caution! You can lose conflicting updates this way.*

#deletedirs <pattern>

tells reconcile always to propagate directory deletions, rather than renaming the deleted directory and generating a warning message.

#nottranslate <pattern>

disables automatic line-end translation in files matching the pattern.

#normal <pattern>

overrides all the other directives, restoring normal handling to files with matching names. Note again that the first matching pattern found is the one used; directives from subdirectories are considered before directives from parent directories in a file hierarchy and directives for a given directory are considered in the order they appear in the .reconcilerc file.

Revision History

1. Initial version, TR93-09, June 1993.
2. Revised and reformatted, TR98-09a, July 1994.
3. Major revisions, TR98-04, April 1998.
4. Revised (additional parameters and directives), TR98-04a, June 1998.